Parallel WAN Switch Based on Neural Network

VLADISLAV SKORPIL, MICHAL POLIVKA
Department of Telecommunications
Brno University of Technology
Purkynova 118, 612 00 Brno
CZECH REPUBLIC
skorpil@feec.vutbr.cz (http://www.utko.feec.vutbr.cz/~skorpil/),
michal.polivka@phd.feec.vutbr.cz (http://www.utko.feec.vutbr.cz/~polivka/)

Abstract: - The article deals with a WAN switch design based on a Feedforward neural network, specifically for the Feedforward Backpropagation algorithm. The designed switch is fully parallel, uses neural network for switch management and also for traffic engineering. The switch uses advanced packet dropping mechanism. The article describes the switch design (network processor design) and compares the developed switch with other "conventional" architectures. Architectures and performance are then compared.

Key-Words: - MATLAB, model, network, neural network, switch, switch algorithm

1 Introduction

Maximal acceptable delay in a converged network is given by desired services – especially in case telephone usage. ITU-T G.114 in section 4 [**1**] determine requests to maximal one-way delay for VoIP in most cases to max 150 ms. The goal of the article and also of the our research is to design and present a new network switch controlled by a neural network, with QoS support and with minimal additional latency [**2**].

1.1 Latency

Latency on each active network component is part of total delay on the transmission path. Total absolute latency is given by equation (1.1), where (l_a) is absolute latency, (l_h) hard latency, (l_v) variable latency, and i is number of latency sources.

$$l_a = \sum_{i=0}^{\infty}\left(l_{h_i} + l_{v_i}\right) \qquad \textbf{(1.1)}$$

Switch is one latency source from many sources in a network. The goal of presented project is to minimize latency on the switch.

A few measurements were carried out and then published in the [**3**]. The measurements were made at the three standard switches and one virtual (VNUML). Standard switches are Micronet SP608K, HP Pro Curve 2626 and Cisco Catalyst 2960. The Micronet switch is designed for SOHO (small office, home office) usage; Catalyst and HP are switches for office usage. For results see fig. 1 and [**3**]. The measured latencies are from 100 to 650 µs. The latencies seem to be deeply under the recommended 150 ms, but the switches are only one latency source. During measurement QoS (Quality of Service) rules were not used. QoS usually adds latency.

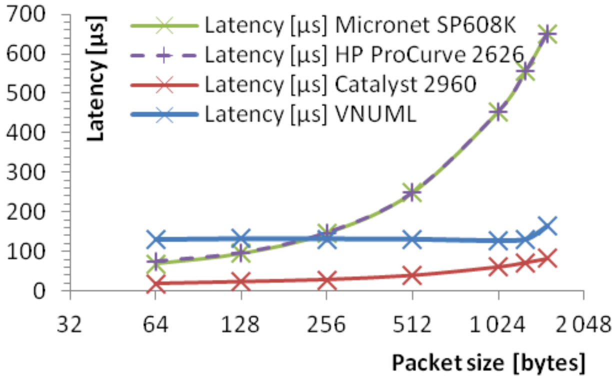

Fig. 1: Switch latency comparison [3]

In some issues presented in fig. 2 some traffic must be prioritized.

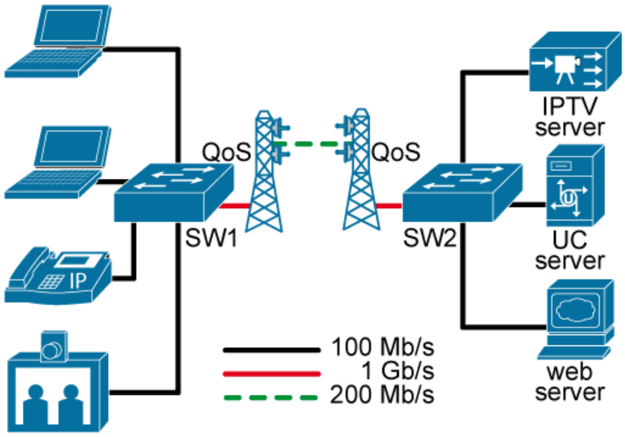

Fig. 2: QoS issue

In fig. 2 2 networks connected by wireless link are presented. The wireless link is the slowest part of the network. The switches SW1 and SW2 in fig. 2 should start using QoS rules. In other case a user experience with services like videoconferencing or VoIP (Voice over IP) will be poor.

The main goal of presented research is to design a switch with QoS support and also with latency exceeding maximally 1 ms.

1.2 Packet Size

The knowledge of packet size distribution is together with protocol distribution important for original QoS setup and also for a bandwidth planning.

For a modeling and test purposes measurement at the laboratory with 12 PC and with standard office protocol usage distribution was carried out. The measurement takes non-stop 24 days and 16 hours. Most of the traffic was protocol http (83 %). The other protocols like RDP, DHCP, ftp, RTP etc. takes remaining 17 %.

For the measurement switch Cisco SGE 2010P was used. In fig. 3 the packet size distribution without pay attention to payload is presented.

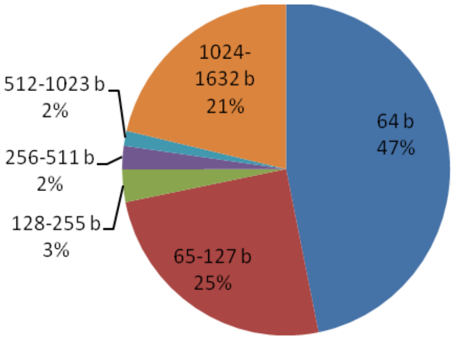

Fig. 3: Packet size distribution – size

The fig. 4 is recalculation of packet distribution with paying attention to payload transmission. The fig. 3 together with fig. 4 show that the packets with size 1024–1632 b take only 21 % of bandwidth but in the packets approximately 78 % of payload is transmitted.

The opposite of the big packets are 64b packets. The 64b packets take 67 % of bandwidth, but they transmit only 8 % of the payload. The measurement does not confirm the packet size 512 b as the most common packet size on the Internet [4]. The packet size distribution depends most of all on the specific network – configuration, users etc., and must be measured independently for each network.

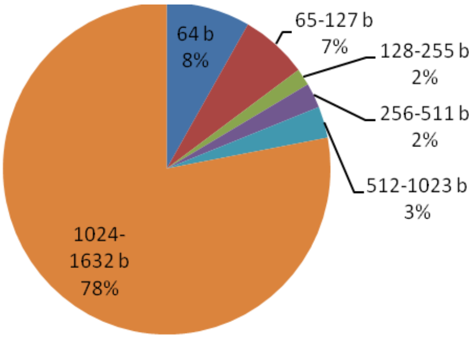

Fig. 4: Packet size distribution – payload

2 Related Work

Switch performance testing and switch modeling is the topic of the article “VNUML – application in computer network, switch modeling” written by Laurent Perroton, Michal Polívka and Tomáš Pelka [3].

The book “High Performance Switches and Routers” written by Jonathan Chao and Bin Liu gives perfect inside to switch design point of issue [5].

The article “Modeling Logical Function Antivalence Using Neural Network in MATLAB” written by Michal Polívka and Vladislav Škorpil deals with modeling XOR function using neural network, but also with neural network selection in specific cases [6].

The book “Network Processors – Architectures, Protocols and Platforms” written by Panos C. Lekkas is one of the best books which deals with network processors design [7].

3 Model

For modeling switch with 4 input/output ports was used. The port number was chosen as performance, transparency and functionality compromise. In the 4 port switch it is possible to test queuing QoS features, observe algorithm errors and still the model is fast and transparent.

QoS support in the model is one of the main goals. At the first phase simplified LLQ (Low Latency Queuing) [8] was chosen for modeling. The LLQ simplification means that simulation does not protect queues with lower precedence. It means that the packet with highest precedence is sent as packet with “strict-priority”.

3.1 Design

The switch model was made after theoretical proposal at the MATLAB software. In the model “Neural Network Toolbox”, some blocks from

"Signal Processing Blockset", basic "Simulink" toolbox and not well known, but useful utility "Real-time" based on script "waitforreal", designed by Stan Quinn [**9**] were used. The pseudo "Real-time" utility is the easiest way to observe the model during running simulation than using full "Real Time Toolbox".

Simulation model is designed as a model working with continual time – it is for maximal model transparency.

Integration method for solving simulation was chosen with "Fixed-Step", and "discrete – no continuous states". Step size is 0.1 – it is compromised selection between accuracy and simulation speed.

Simulation usually starts at time 0 and finishes at time 20. The model is flexible and the simulation time can be adjusted if necessary.

The model at a top layer consists of 4 main kinds of blocks – packet generator, input queues, switch fabric and output FIFOs. For schema see fig. 5.

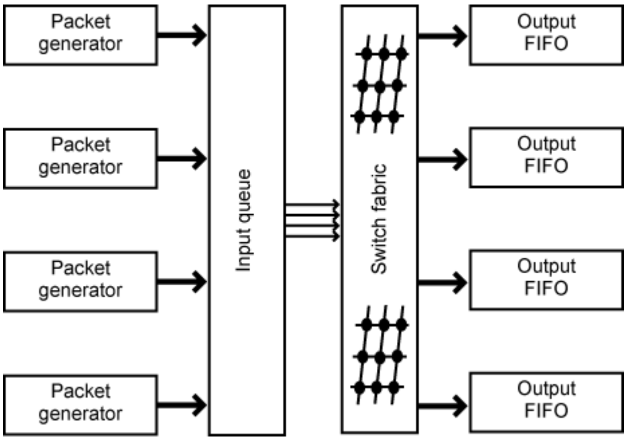

Fig. 5: Switch schema [10]

3.2 Test Packet

Special test packet was developed for the simulation model. The packet is inspired by IPv4 packet structure [**11**], but the structure is thus extremely simplified. Structure of the developed packet is in fig. 6.

0	1	2	3	4	5
Source Port	Destination Port	Priority		Data	

Fig. 6: Packet design [10]

Packet consists of 4 arrays – source and destination ports, priority and data array.

- **Source port** – identifies the packet source – port (generator), see section 3.3. Source port is whole number 1–4.
- **Destination port** – identifies the packet destination port. Destination port is whole number 1–4.
- **Priority** – the array defines the packet priority. The priority number consists of 2 parts. The priority array assumes DSCP tagging, see RFC 2597 [**12**]. First part of the number is whole number 1–4, and it represents priority (P). Higher number is higher priority. Second part of number is whole number 1–3 and represents a probability of packet drop (d). The higher value means higher drop probability. Calculation of array value is given by equation (3.1).

$$priority = P \cdot 10 + d \qquad \textbf{(3.1)}$$

- **Data** – array represents the "real" transferred data. The array is whole number 1–10,000 and in binary representation it is a number with variable length. Because the data array is usually random number and the probability of repetition in short time interval is low, the array is also used for packet identification during checking the packet processing.

The test packet is designed for easy conversion to input vector for neural network, see section 3.5.1.

3.3 Packet Generator

The model contains 4 identical packet generators. Generated sequence can be "random" or user defined. Each packet generator is controlled independently, and also each array of generated packet is independently configurable. In random mode Gaussian distribution of probability or uniform distribution probability can be used. The output block in the generator consists of FIFO (First In, First Out) queues. The FIFOs are read in random time – it is simulation of jitter in a network.

3.4 Input Queue

In the model there is the block "input queue" that is responsible for packet scheduling. All packets generated by generators are sent to the input queues. The input queue block contains the collision detector sub-block. If one or more packets have identical destination there are 2 possibilities of behavior.

The first possibility of behavior, when packet destination collides, is simply drop the packet with lower priority (P). In case of collision of more than 2 packets, packet with higher priority proceeds.

The second possibility is much more sophisticated. Inbuilt sub-blocks support QoS. Two or more packets in collision are sent to QoS processing sub-block. QoS sub-blocks are divided to the groups. For each output port one queue for 8 packets is defined. Each main queue consists of additional queues – for the each precedence is defined independent queue.

In case the queue is full (the queues are designed for 8 packets), new incoming packets are dropped. This algorithm is known as “tail drop” with “aggressive dropping” [**8**], [**13**].

In case the switch fabric is ready for packet processing, the packets with higher precedence are read first.

3.5 Switch Fabric

Switch fabric is modeled as 4×4 array. The array consists of matrix of simple controlled switches. The switches are controlled by configuration matrix (c), for example see equation (3.2).

$$c = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad \textbf{(3.2)}$$

The configuration matrix is generated by control neural network. The switch is designed for parallel packet processing. In ideal condition it can control the increasing port number without decreasing performance.

3.5.1 Neural Network

The configuration matrix is generated by “Feedforward Backpropagation” neural network. Input vector of neural network is packet converted to vector, output of network is configuration matrix. In hidden layer 100 neurons are used. Output layer consists of 4 neurons. For visualization see fig. 7.

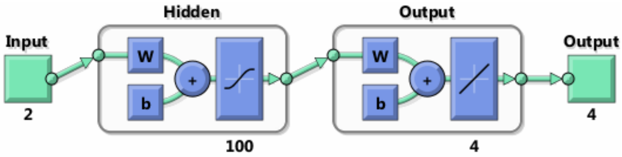

Fig. 7: Neural network visualisation [10]

The designed neural network converged to demanded mean squared error (mse) quickly. After 3 epochs $mse = 4{,}306 \cdot 10^{-21}$ was reached. The fitting function is in fig. 8.

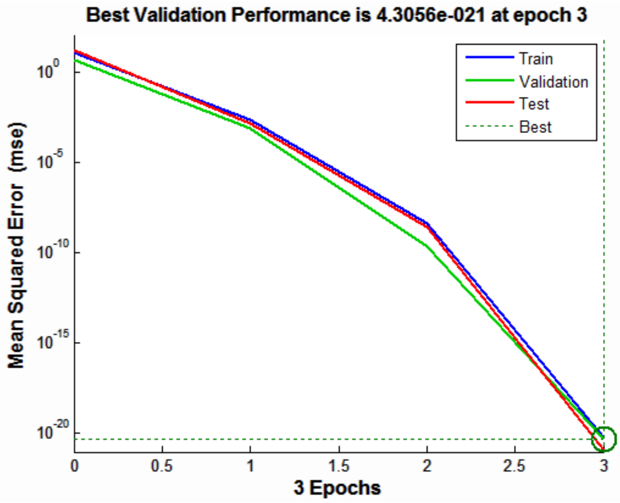

Fig. 8: Neural network fitting performance [10]

Mean squared error is defined by (3.3), [**14**].

$$mse = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - a_i)^2 \qquad \textbf{(3.3)}$$

The t_i in (3.3) is set of target values during network fitting, and a_i is set of real network outputs.

3.6 Output FIFO

For each output port one output FIFO queue is defined. The queue size is set to save 8 packets – same size as is set for input queues. The packets are saved to output queues and they are sent immediately when opposite device is ready for receiving the packet. In the model it is simulated by pseudorandom generator which allows to send packet in random time.

4 Conclusion

The present model is a part of complex QoS research. The model verified viability of switch based on the neural network.

The research will continue with model conversion to the VHDL language and modeling at a FPGA. The FPGA based model allow to compare switch performance with other formerly measured (fig. 1) switches in a real environment.

Acknowledgement

This research was supported by the grants: No CZ.1.07/2.2.00/28.0062, Joint activities of BUT and TUO while creating the content of accredited technical courses in ICT, No. ED 2.1.00/03.0072, Centre of sensor information and communication systems (SIX) and No. FEKT-S-11-15 Research of electronic communication systems.

References:

[1] International Telecommunication Union. (2003, May) G.114: One-way transmission time. [Online]. http://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-G.114-200305-I!PDF-E&type=items

[2] Kevin Wallace, *Cisco VoIP*. Brno: Computer Press, a. s., 2009.

[3] Michal Polívka and Laurent Perroton, "VNUML – application in computer network switch modeling," 2009.

[4] Cisco Systems Inc. (2007, Dec.) IP to ATM Class of Service. [Online]. http://www.cisco.com/en/US/tech/tk39/tk824/technologies_tech_note09186a00800fbafc.shtml

[5] Jonathan H. Chao and Bin Liu, *High Performance Switches and Routers*. New Jersey: JohnWiley & Sons, Inc., 2007.

[6] Michal Polívka and Vladislav Škorpil, "Modeling Logical Function Antivalence Using Neural Network in MATLAB," in *13th International Conference on Research in Telecommunication Technologies RTT - 2011*, Brno, 2011, pp. 1-3.

[7] Panos C. Lekkas, *Network Processors – Architectures, Protocols and Platforms*. New York: McGraw-Hill, 2003.

[8] Wendell Odom, Rus Healy, and Naren Mehta, *Směrování a přepínání sítí – Autorizovaný výukový průvodce*. Brno: Computer Press, a.s.; Cisco System, Inc., 2009.

[9] The MathWorks, Inc. (2010, July) How do I get a real time clock in simulations using Simulink? [Online]. http://www.mathworks.com/support/solutions/en/data/1-15JAW/

[10] Michal Polívka and Vladislav Škorpil, "Modeling of Network Switch Controlled by Neural Network," in *35th International Conference on Telecommunications and Signal Processing (TSP),* Prague, 2012.

[11] Information Sciences Institute, University of Southern California. (1981, Sep.) RFC 791: Internet Protocol Darpa Internet Program Protocol Specification. [Online]. http://datatracker.ietf.org/doc/rfc791/

[12] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. (1999, June) RFC 2597: Assured Forwarding PHB Group. [Online]. http://datatracker.ietf.org/doc/rfc2597/

[13] Wendell Odom and Michael J. Cavanaugh, *Cisco QoS Exam Certification Guide (IP Telephony Self-Study), 2nd Edition*. Indianapolis: Cisco Press, 2004.

[14] Mark Hudson Beale, Martin T. Hagan, and Howard B. Demuth, *Neural Network Toolbox 7*. Natic: The MathWorks, Inc., 2010.